Model Checking for Verification of Quantum Circuits

Mingsheng Ying[1,2,3](✉)

[1] Centre for Quantum Software and Information, University of Technology Sydney, Ultimo, Australia
Mingsheng.Ying@uts.edu.au
[2] State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China
[3] Department of Computer Science and Technology, Tsinghua University, Beijing, China

Abstract. In this survey paper, we describe a framework for *assertion-based verification* of quantum circuits by applying *model checking* techniques for quantum systems developed in our previous work, in which:

- Noiseless and noisy quantum circuits are *modelled* as operator- and super-operator-valued transition systems, respectively, both of which can be further represented by tensor networks.
- Quantum *assertions* are specified by a temporal extension of Birkhoff-von Neumann quantum logic. Their semantics is defined based on the following design decision: they will be used in verification of quantum circuits by simulation on classical computers or human reasoning rather than by quantum physics experiments (e.g. testing through measurements);
- *Algorithms* for reachability analysis and model checking of quantum circuits are developed based on contraction of tensor networks. We observe that many optimisation techniques for computing relational products used in BDD-based model checking algorithms can be generalised for contracting tensor networks of quantum circuits.

Keywords: Quantum logic circuits · Verification · Assertion · Temporal logic · Model checking · Reachability · Tensor network

1 Introduction

Assertion-based verification (ABV) is a key methodology for functional verification of classical logic circuits and has been widely adopted in hardware industry. A major characteristic of ABV is that assertions are used for specifying design intent at a high level of abstraction and thus are ideal for using across multiple verification processes [10]. An example application procedure of ABV was described in [4] as follows:

M. Huisman et al. (Eds.): FM 2021, LNCS 13047, pp. 23–39, 2021.
https://doi.org/10.1007/978-3-030-90870-6_2

1. A specification language such as PSL (Property Specification Language, IEEE 1850 standard) or SVA (SystemVerilog Assertions) is used to write the *assertions* specifying the desired hardware properties.
2. Verification is performed by formal methods or in a dynamic manner where a *simulator* monitors the device under verification (DUV) and reports when and where assertions are violated.
3. The information on assertion violation can be used in the *debugging* process.

Verification of quantum circuits is emerging as an important issue duo to the rapid growth in the size of quantum computing hardware. A majority of the current research has been devoted to *equivalence checking* of *combinational* quantum circuits using various quantum generalisations of BDDs (Binary Decision Diagrams), such as QuIDD [24,25], and QMDD [6,22]. Recently, *sequential* circuit models are emerging to play an important role in quantum computing and information processing; examples include quantum memories [18], quantum feedback networks [12], and RUS (Repeat-Until-Success) quantum circuits [3]. A hardware description language was defined in [23] for specification of sequential quantum photonic circuits. An algorithm for equivalence checking of sequential quantum circuits is presented in [28]. One can expect that as more and more sophisticated quantum hardware be physically realisable, more and more complicated verification problems will appear for quantum circuits, and assertion-based verification (ABV) will become an indispensable technology in future design automation for quantum computing (QDA).

Model Checking Quantum Systems: Essentially, assertion-based verification (ABV) of logic circuits can be seen as an important application of temporal logic and model checking. Research on extending model checking for quantum system has been conducted in the last fifteen years. Early work aimed at verification of quantum communication protocols [1,8,11]. Targeting applications in analysis and verification of quantum programs [29], several model checking techniques for quantum automata, quantum Markov chains and super-operator valued Markov chains have been developed in [9,20,33,36] (see [31] for a more systematic exposition). However, a big gap between these quantum model checking techniques and their practical applications in verification of quantum circuits is still to be filled in. For near term applications, we believe that the following two challenges are crucial:

- ***Challenge I*** - *Finding compact representations of quantum circuits*: As a compact representation, BDDs have played a key role in successful applications of model checking in verification of classical circuits [5]. As pointed out before, several quantum generalisations of BDDs have been employed in equivalence checking of quantum circuits. On the other hand, tensor networks - a mathematical tool successfully applied in simulation of quantum physical system for decades - have been widely used in simulation of large quantum circuits on classical computers in the last few years [15,17,19,21,27]. These representations should be helpful in implementing a more efficient model checker for quantum circuits.

- ***Challenge II** - Identifying useful properties that can be checked by the current technology*: The previous research pursued theoretical generality and thus targeted checking general reachability and temporal logic properties of quantum systems. But a model checker (implemented on a classical computer) for such a purpose must be highly inefficient and only applicable to quantum circuits of very small sizes and depths. Thus, for realistic and in particular, near-term applications, we need to identify a class of simpler properties that can be efficiently checked by a current model checker for quantum systems.

In this survey paper, we describe a framework for *assertion-based verification* (ABV) of quantum circuits by applying *model checking* techniques for quantum systems developed in our previous work, in which quantum circuits are represented by tensor networks, and assertions about quantum circuits are specified using a simple temporal extension of Birkhoff-von Neumann quantum logic. The paper is organised as follows. Basic models of quantum circuits are reviewed in Sect. 2. To address Challenge I, we introduce tensor network representation of quantum circuits in Sect. 3. The reason for using tensor networks is that algorithms for reachability analysis and model checking of quantum circuits can be conveniently implemented by contraction of tensor networks. More importantly, we observe that many optimisation techniques for computing relational products used in BDD-based model checking algorithms can be generalised for contracting tensor networks of quantum circuits. Challenge II is gradually addressed in Sects. 4 to 6. In Sect. 4, we first show how a basic property, namely reachability, of quantum circuits can be checked. To specify more general properties, a simple temporal logic is defined in Sect. 5 as our assertion language. This language is chosen because it is actually useful for practical applications and at the same time, checking assertions written in it is much easier than for other assertion languages, as shown in Sect. 6. Furthermore, its semantics will be defined based on the following design decision: the target application is verification of quantum circuits by simulation on classical computers (or human reasoning) rather than by quantum physics experiments (e.g. testing through measurements). We hope that focusing on this more realistic target, a model checker can be built for practical use in verification and debugging of near term quantum hardware.

2 Quantum Logic Circuits

For convenience of the audience, let us start from a brief review of the basics of quantum computing, with the emphasis on several basic models of quantum circuits.

2.1 Combinational Quantum Circuits

Traditional combinational circuits are made from logic gates acting on wires. Combinational quantum circuits are quantum counterparts of them and made up of quantum (logic) gates, which are modelled by unitary operators.

Qubits (Quantum Bits): The quantum counterpart of a bit is a qubit. A state of a single qubit is represented by a 2-dimensional unit column vector $(\alpha, \beta)^T$, where T stands for transpose, and complex numbers α, β satisfy the normalisation condition $\|\alpha\|^2 + \|\beta\|^2 = 1$. It can be conveniently written in the Dirac's notation as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ with $|0\rangle = (1,0)^T$, $|1\rangle = (0,1)^T$ corresponding to classical bits 0 and 1, respectively. Intuitively, this qubit is in a superposition of 0 and 1. In general, we use $q, q_1, q_2, ...$ to denote qubit variables. Graphically, they can be thought of as wires in a quantum circuit. A state of n qubits $q_1, ..., q_n$ is then written as a 2^n-dimensional unit complex vector $(\alpha_0, \alpha_1, ..., \alpha_{2^n-1})^T$ or in the Dirac's notation:

$$|\psi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x|x\rangle = \sum_{x_1,\dots,x_n} \alpha_{x_1,\dots,x_n}|x_1, ..., x_n\rangle \tag{1}$$

where its norm $\||\psi\rangle\| = \sqrt{\sum_x |\alpha_x|^2} = 1$, and we exchangeably use an n-bit string $x = x_1...x_n \in \{0,1\}^n$ and integer $x = \sum_{i=1}^n x_i \cdot 2^{i-1}$.

Quantum Gates: A gate on a single qubit is modelled by a 2×2 complex matrix U. In general, a gate on n qubits is described by a $2^n \times 2^n$ unitary matrix

$$U = \left(U_{x,y}\right)_{x,y\in\{0,1\}^n}. \tag{2}$$

The output of U on an input $|\psi\rangle$ is quantum state $|\psi'\rangle$. Its mathematical representation as a vector is obtained by standard matrix multiplication $|\psi'\rangle = U|\psi\rangle$. To guarantee that $|\psi'\rangle$ is always unit, U must be unitary in the sense that $U^\dagger U = I$, where $U^\dagger$ is the adjoint of U obtained by transposing and then complex conjugating U. We often write $G \equiv U[q_1, ..., q_n]$ for gate U acting on qubits $q_1, ..., q_n$.

Example 1. *1. The following are several frequently used single-qubit gates:*

(a) Hadamard gate: $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$;

(b) The Pauli matrices: $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, $Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$, $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$.

2. Let q_1, q_2 be qubits. Then CNOT (controlled-X) gate $C[q_1, q_2]$ is a two-qubit gate with q_1 as the control qubit and q_2 as the target qubit and defined by the 4×4 matrix $C = \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix}$, where I is the 2×2 identity matrix.

Combinational Quantum Circuits: A combinational quantum circuit is a sequence of quantum gates: $C \equiv G_1...G_m$, where $m \geq 1$ and $G_1, ..., G_m$ are quantum gates.

Example 2. *The quantum circuit $Z[q_1]H[q_2]C[q_1, q_2]Y[q_1]H[q_2]$ consisting of five quantum gates is visualised in Fig. 1.*

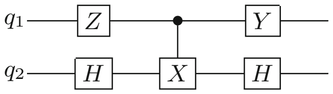

Fig. 1. A combinational quantum circuit.

2.2 Noisy Quantum Circuits

Fault-tolerant quantum computing is still out of the current technology's reach. To model noisy implementation of quantum circuits, we recall that a mixed state of an n-qubit system is an *ensemble* $\{(|\psi_i\rangle, p_i)\}$ of its pure states, meaning that this system is in state $|\psi_i\rangle$ with probability p_i. Mathematically, this mixed state can be described by a $2^n \times 2^n$ matrix, called a density matrix, $\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$, where $\langle\psi_i|$ is the conjugate transpose of $|\psi_i\rangle$ and thus a 2^n-dimensional row vector. In particular, a pure state $|\psi\rangle$ can be identified with the outer product $|\psi\rangle\langle\psi|$. Then a noisy n-qubit gate can be modelled by a super-operator, often called a quantum channel in quantum information literature, which is a linear map $\mathcal{E} : \rho \to \mathcal{E}(\rho)$ from $2^n \times 2^n$ density matrices to themselves. A convenient way of representing $\mathcal{E}$ is the *Kraus operator-sum form*:

$$\mathcal{E}(\rho) = \sum_i E_i \rho E_i^\dagger \tag{3}$$

for any density matrix ρ, where $\{E_i\}$ is a set of $2^n \times 2^n$ matrices satisfying the normalisation condition $\sum_i E_i^\dagger E_i = I_{2^n}$. In particular, an idea n-qubit gate modelled by a unitary operator U can be seen as a super-operator $\mathcal{U} : \rho \mapsto U\rho U^\dagger$.

Example 3. *Several canonical noises on a single qubit are:*

1. *Bit flip: This noise flips the state of a qubit from $|0\rangle$ to $|1\rangle$ and vice versa with probability $1-p$, and is modelled by super-operator $\mathcal{N}_{bf}(\rho) = p\rho + (1-p)X\rho X$.*
2. *Phase flip: This noise changes the phase of a qubit (that is, applies phase operator Z on the qubit) with probability $1-p$, and is modelled by the super-operator $\mathcal{N}_{pf}(\rho) = p\rho + (1-p)Z\rho Z$.*
3. *Bit-phase flip: This noise applies Pauli operator Y on a qubit with probability $1-p$: $\mathcal{N}_{bpf}(\rho) = p\rho + (1-p)Y\rho Y$. Note that it is essentially a combination of a bit-flip and a phase flip because $Y = iXZ$.*

2.3 Dynamic Quantum Circuits

Quantum Measurement: The output of a combinational quantum circuit is a quantum state, which cannot be observed directly from the outside. To read out the outcome of computation, we have to perform a measurement at the end of the circuit. Mathematically, a quantum measurement on n qubits is described by a family $M = \{M_m\}$ of $2^n \times 2^n$ matrices such that $\sum_m M_m^\dagger M_m = I_{2^n}$, where m

denotes different possible outcomes. If one performs M on the qubits in state $|\psi\rangle$, then outcome m is obtained with probability $p_m = \| M_m|\psi\rangle\|^2$ and subsequently the state of these qubits will be changed to $\frac{M_m|\psi\rangle}{\sqrt{p_m}}$. More generally, if the n-qubit system is in a mixed state ρ, then outcome m is obtained with probability $p_m = \mathrm{tr}(M_m^\dagger M_m \rho)$ and its state will be changed to $\frac{M_m \rho M_m^\dagger}{p_m}$. For example, the measurement in the computational basis is defined as $M = \{M_x : x \in \{0,1\}^n\}$ with $M_x = |x\rangle\langle x|$, and if it is performed on the qubits in a pure state (1), then outcome $x \in \{0,1\}^n$ is obtained with probability $|\alpha_x|^2$ and subsequently the qubits will be in basis state $|x\rangle$.

Dynamic Quantum Circuits: Quantum measurements are not only used for readout of the computational outcome at the end of a quantum circuit as described above. They may also occur at the middle of a quantum circuit where the measurement outcomes are used to conditionally control subsequent steps of the computation. This kind of circuits are called dynamic quantum circuits [7] and have been realised in several hardware platforms for quantum computing. Formally, they are inductively defined as follows (see [29], p. 38):

- (Noiseless or noisy) quantum gates are dynamic quantum circuits;
- If C_1, C_2 are dynamic quantum circuits, so is $C_1; C_2$; and
- If $M = \{M_m\}$ a measurement on qubits $q_1, ..., q_n$, and for each possible outcome m, C_m a dynamic quantum circuit , then **if** $(\Box m \cdot M[q_1, ..., q_n] = m \rightarrow C_m)$ **fi** is a dynamic quantum circuit. Intuitively, this conditional circuit performs measurement M on qubits $q_1, ..., q_n$, and then the subsequent computation is selected based on the measurement outcome: if the outcome is m, then the corresponding circuit C_m follows.

Quantum teleportation is a simple example of dynamic quantum circuits. Another example is the dynamic circuit for quantum phase estimation shown as Fig. 1 in [7].

Example 4. *Quantum teleportation is a protocol for transmitting quantum information (e.g. the exact state of an atom or photon) via only classical communication but with the help of previously shared quantum entanglement between the sender and receiver. It is one of the most surprising examples where entanglement helps to accomplish a certain task that is impossible in the classical world. The quantum circuit teleporting a single qubit is shown in Fig. 2.*

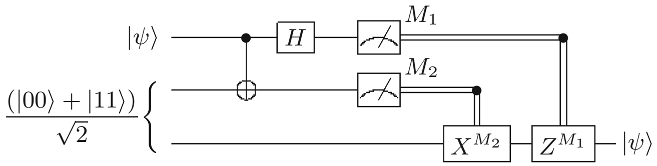

Fig. 2. Quantum teleportation circuit

2.4 Sequential Quantum Circuits

As is well-known, almost all practical digital devices contain (classical) sequential circuits. The output value of a combinational circuit is a function of only the current input value. In contrast, the output value of a sequential circuit depends on not only the external input value but also on the stored internal information. All quantum circuits considered in the previous subsections are combinational. However, several recent applications introduce a *sequential* model of quantum circuits, including quantum memories [18], quantum feedback networks [12], and RUS (Repeat-Until-Success) quantum circuits [3]. A synchronous model of sequential quantum circuit was defined in [28] and can be visualised as Fig. 3, which looks similar to its classical counterpart, except:

- The combinational part of a classical sequential circuit is modelled by a Boolean function; whereas the combinational part of a sequential quantum circuit is a unitary operator or a super-operator, depending on whether noise occurs in it.
- Certain measurements are needed at the end of qubits $q_1, ..., q_k$ to readout classical information from their outputs.

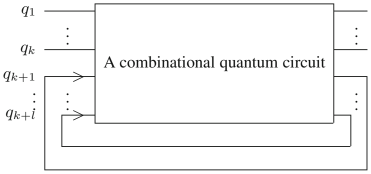

Fig. 3. A sequential quantum circuit

2.5 Quantum Transition Systems as a Model of Quantum Circuits

A classical circuit can be conveniently described by a transition relation [5]. Similarly, quantum circuits discussed above can be modelled by a quantum transition system defined as follows.

Definition 1 (Quantum Transition Systems). *A quantum transition system (QTS) for a circuit with n qubits consists of:*

1. *a finite set L of locations, and an initial location $l_0 \in L$;*
2. *a set $\mathcal{T}$ of transitions:*
   - *each transition $\tau \in \mathcal{T}$ is a triple $\tau = \langle l, l', \mathcal{E} \rangle$, often written as $\tau = l \xrightarrow{\mathcal{E}} l'$ where $l, l' \in L$ are the pre- and post-locations of τ, respectively, and $\mathcal{E}$ is a super-operator on $2^n \times 2^n$ density matrices,*

   satisfying the normalisation condition: $\sum\{|\mathrm{tr}[\mathcal{E}(\rho)] : l \xrightarrow{\mathcal{E}} l' \in \mathcal{T}|\} = 1$ for each $l \in L$ and $2^n \times 2^n$ density matrix ρ, where $\{|\cdot|\}$ stands for a multi-set, and trace $\mathrm{tr}(A)$ of a matrix A is the sum of the entries on the diagonal of A.

In particular, for a noiseless quantum circuit, every transition $l \xrightarrow{\mathcal{E}} l'$ is simply defined by a $2^n \times 2^n$ matrix E such that $\mathcal{E}(\rho) = E\rho E^\dagger$ for all density matrices ρ; for example, each quantum gate is defined as a unitary matrix U, and in a quantum measurement $M = \{M_m\}$, each branch corresponding to an outcome m can be described by the measurement operator M_m.

Example 5. *The circuit of quantum teleportation in Fig. 2 can be modelled by the QTS in Fig. 4, where quantum operations are visualised by edges; for example, $CX_{1,2}$ on edge $l_0 \to l_1$ denotes a CNOT on qubits 1 and 2, and $M_{2,1}$ on edge $l_2 \to l_4$ means that a measurement is performed on qubit 2 and outcome 1 is obtained.*

Remark 1. QTS's were first introduced in [9,14] where they are called quantum Markov chains. They were also used in defining invariants of quantum programs [34].

3 Tensor Network Representation of Quantum Circuits

In the last section, quantum circuits were defined in the traditional vector and matrix language of quantum mechanics. The shift from representing quantum circuits by matrices to tensor networks was proposed in [21] by identifying the following benefits in simulation of quantum circuits on classical computers: (i) quantum circuits can be arbitrarily partitioned into subcircuits; (ii) subcircuits can be simulated in arbitrary orders; and (iii) simulation results of subcircuits can be combined in arbitrary orders. From these benefits, the reader might already notice that the advantage of tensor network representation of quantum circuits over matrices is very much similar to that of BDD representation of classical circuits over truth tables (i.e. Boolean matrices). In this section, we briefly review the basic idea of tensor networks and show how they can be used to represent quantum circuits.

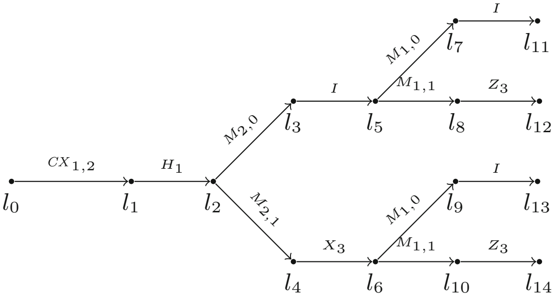

Fig. 4. A QST for quantum teleportation

3.1 Tensor Networks

A tensor is a multi-dimensional array of complex numbers. We only consider a special class of tensors suitable for representing quantum circuits. A tensor with an index set $\vec{q} = \{q_1, ..., q_n\}$ is a mapping $T : \{0,1\}^{\vec{q}} \to \mathbb{C}$. We often write $T = T_{\vec{q}}$ or $T_{q_1,...,q_n}$ to indicate the indices. For two tensors $T_{\vec{p},\vec{r}}$ and $T_{\vec{q},\vec{r}}$ sharing indices $\vec{r}$, their *contraction* is defined as a tensor $T_{\vec{p},\vec{q}} \triangleq \mathit{Contract}(T_{\vec{p},\vec{r}}, T_{\vec{q},\vec{r}})$ by

$$T_{\vec{p},\vec{q}}(\vec{a},\vec{b}) = \sum_{\vec{c}\in\{0,1\}^{\vec{r}}} T_{\vec{p},\vec{r}}(\vec{a},\vec{c}) \cdot T_{\vec{q},\vec{r}}(\vec{b},\vec{c}) \tag{4}$$

for any $\vec{a} \in \{0,1\}^{\vec{p}}$ and $\vec{b} \in \{0,1\}^{\vec{q}}$. Then a tensor network is a hyper-graph $H = (V, E)$, where a subset $E_0 \subseteq E$ is chosen as open edges, and each vertex $v \in V$ is associated with a tensor of which the hyper-edges incident to v are the indices. Thus, the hyper-edges between two vertices represent the indices shared by the two adjacent tensors. By contracting connected tensors in H, we can obtain a tensor T_H with E_0 as its index set. It is easy to see that T_H is independent of the order of contractions.

3.2 Representing Quantum States and Quantum Gates

The tensor representation of quantum states is straightforward. A pure state $|\psi\rangle$ of n qubits $q_1, ..., q_n$ given in Eq. (1) can be represented by a tensor $T_{|\psi\rangle} \triangleq T_{q_1,...,q_n}$ with $T_{q_1,...,q_n}(x_1,...,x_n) = \alpha_{x_1,...,x_n}$ for any $x_1,...,x_n \in \{0,1\}$. Furthermore, a mixed state of n qubits $q_1, ..., q_n$ given as a $2^n \times 2^n$ density matrix $\rho = (\rho_{x,y})_{x,y\in\{0,1\}^n}$ can be represented by a tensor $T_\rho \triangleq T_{q_1,...,q_n,q_1',...,q_n'}$, where for any $x, y \in \{0,1\}^n$:

$$T_{q_1,...,q_n,q_1',...,q_n'}(x,y) = \rho_{x,y}. \tag{5}$$

Similar to the tensor representation (5) of a density matrix, a (noiseless) quantum gate U on n qubits $q_1, ..., q_n$ given as unitary matrix (2) can be straightforwardly represented by a tensor $T_U \triangleq T_{q_1,...,q_n,q_1',...,q_n'}$ with $T_{q_1,...,q_n,q_1',...,q_n'}(x,y) = U_{x,y}$ for any $x, y \in \{0,1\}^n$. To present a tensor representation of a noisy quantum gate $\mathcal{E}$ on n qubits $q_1, ..., q_n$, we assume that it is given in the Kraus representation (3), and define its matrix representation as

$$M_{\mathcal{E}} = \sum_i E_i \otimes E_i^* \triangleq (M_{x,y,x',y'})_{x,y,x',y'\in\{0,1\}^n} \tag{6}$$

where E^* stands for the conjugate of E; that is, if $E = (E_{x,y})$, then $E^* = \left(E^*_{x,y}\right)$, and $E^*_{x,y}$ is the conjugate of complex number $E_{x,y}$ for any $x, y \in \{0,1\}^n$. Furthermore, if for each qubit q_i, we introduce a new copy p_i, then $M_{\mathcal{E}}$ can be represented by a tensor $T_{\mathcal{E}} \triangleq T_{q_1,...,q_n,p_1,...,p_n,q_1',...,q_n',p_1',...,p_n'}$, where for any $x, y, x', y' \in \{0,1\}^n$:

$$T_{q_1,...,q_n,p_1,...,p_n,q_1',...,q_n',p_1',...,p_n'}(x,y,x',y') = M_{x,y,x',y'}. \tag{7}$$

3.3 Representing Quantum Circuits

Now we can present a tensor network representation of quantum circuits by assembling the ingredients given in the previous subsections. Suppose we are given a combinational or sequential quantum circuit C modelled as a quantum transition system. If we replace each (noiseless or noisy) gate in C by its tensor representation, then we obtain a tensor network representation of C. Furthermore, one can compute its tensor T_C by contraction (4). Moreover, if $|\psi\rangle$ or ρ is an input to C, then the tensor representation of output $C|\psi\rangle$ or $C(\rho)$ can be computed as contraction $Contract(T_{|\psi\rangle}, T_C)$ or $Contract(T_\rho, T_C)$, respectively. When computing the tensor of a noisy quantum circuit, it is often more efficient to use contraction in combination with the following lemma, which gives a way for computing the matrix representations of the sequential and parallel compositions of noisy quantum gates: the matrix representation of the sequential (respectively, parallel) composition of two noisy quantum gates is the multiplication (respectively, tensor product) of their matrix representation.

Lemma 1. *For any super-operators $\mathcal{E}$ and $\mathcal{F}$, we have $M_{\mathcal{E}\circ\mathcal{F}} = M_{\mathcal{F}}M_{\mathcal{E}}$ and $M_{\mathcal{E}\otimes\mathcal{F}} = M_{\mathcal{F}} \otimes M_{\mathcal{E}}$.*

3.4 Optimisations for Tensor Network Contraction

It is obvious that computation required in the contraction of tensor networks of quantum circuits tends to be exponential in the growth of the number of qubits and the depth of circuits. In the last few years, many optimisation techniques have been proposed in the tensor network-based algorithms for simulation of quantum circuits on classical computers [15,17,19,21,27]. The main reason for employing tensor networks rather than large matrices in simulation of quantum circuits is that tensor networks can exploit the regularity and locality in the structure of quantum circuits. Essentially, the basic idea is similar to that of optimisation strategies in BDD-based algorithms (although this similarity has not been explicitly pointed out in the literature). We believe that more BDD-optimisations can be adapted to computing tensor networks of quantum circuits, in particular when combined with their QST representations defined in Subsect. 2.5; for example, Lemma 1 enables us to generalise the partitioning technique in verification of classical circuits (see [5], Sect. V) to the case of noisy quantum circuits. For this purpose, we introduced a decision-diagram style data structure, called TDD (Tensor Decision Diagram), and showed that various operations of tensor networks essential in their applications can be conveniently implemented in TDDs [16].

4 Reachability Analysis of Quantum Circuits

We now start to consider the verification problem of quantum circuits. Many model checking problems about classical circuits (and other systems) can be reduced to a reachability problem. Reachability plays a similar role in model checking quantum systems [31]. In this section, as a basis of verification techniques for quantum circuits, let us focus on reachability of a simplest version of quantum transition systems, namely a *quantum Markov chain* [35], which is defined as a pair $\mathcal{C} = \langle \mathcal{H}, \mathcal{E} \rangle$, where $\mathcal{H}$ is a finite-dimensional Hilbert space as the system's state space, and $\mathcal{E}$ is a quantum operation (or super-operator) in $\mathcal{H}$ depicting transition of the system's state. Roughly speaking, if the initial state is ρ, then the quantum Markov chain behaves as follows: $\rho \to \mathcal{E}(\rho) \to \cdots \to \mathcal{E}^n(\rho) \to \mathcal{E}^{n+1}(\rho) \to \cdots$.

4.1 Adjacency and Reachability

As in the classical case, a graph structure is helpful for reachability analysis in quantum Markov chain $\mathcal{C}$. Let us first recall several notations needed to define such a graph structure. For any $X \subseteq \mathcal{H}$, let span(X) stand for the subspace spanned by X, i.e. the smallest subspace of $\mathcal{H}$ containing Y. The support supp(A) of an operator A on $\mathcal{H}$ is the subspace spanned by the eigenvectors of A associated with non-zero eigenvalues. For a family $\{X_i\}$ of subspaces of $\mathcal{H}_i$, we define their join as

$$\bigvee_i X_i = \mathrm{span}\left(\bigcup_i X_i\right). \tag{8}$$

In particular, we write $X_1 \vee X_2$ for the join of two subspaces X_1 and X_2. The image of a subspace X of $\mathcal{H}$ under $\mathcal{E}$ is defined as $\mathcal{E}(X) = \bigvee_{|\psi\rangle \in X} \mathrm{supp}(\mathcal{E}(|\psi\rangle\langle\psi|))$, where $|\psi\rangle\langle\psi|$ is the density operator corresponding to pure state $|\psi\rangle$.

Definition 2 (Adjacency Relation). *Let $|\varphi\rangle, |\psi\rangle \in \mathcal{H}$ be pure states and ρ, σ be mixed states (i.e. density matrices) in $\mathcal{H}$. Then*

1. *$|\varphi\rangle$ is adjacent to $|\psi\rangle$ in $\mathcal{C}$, written $|\psi\rangle \to |\varphi\rangle$, if $|\varphi\rangle \in \mathrm{supp}(\mathcal{E}(|\psi\rangle\langle\psi|))$.*
2. *$|\varphi\rangle$ is adjacent to ρ, written $\rho \to |\varphi\rangle$, if $|\varphi\rangle \in \mathcal{E}(\mathrm{supp}(\rho))$.*
3. *σ is adjacent to ρ, written $\rho \to \sigma$, if $\mathrm{supp}(\sigma) \subseteq \mathcal{E}(\mathrm{supp}(\rho))$.*

Then as in classical graph theory, a path from a state ρ to a state σ in $\mathcal{C}$ is a sequence $\rho_0 \to \rho_1 \to \cdots \to \rho_n$ $(n \geq 0)$ of adjacent states such that $\rho_0 = \rho$ and $\rho_n = \sigma$. For any two states ρ and σ, if there is a path from ρ to σ then we say that σ is reachable from ρ in $\mathcal{C}$.

Definition 3 (Reachable Subspace). *For any state ρ in $\mathcal{H}$, its reachable space in $\mathcal{C}$ is the subspace of $\mathcal{H}$ spanned by the states reachable from ρ:*

$$\mathcal{R}_{\mathcal{C}}(\rho) = \mathrm{span}\{|\psi\rangle \in \mathcal{H} : |\psi\rangle \text{ is reachable from } \rho \text{ in } \mathcal{C}\}.$$

The following theorem from [37] gives a useful characterisation of reachable subspaces. It is essentially a generalisation of Kleene closure in relational algebra.

Theorem 1. *Let $d = \dim \mathcal{H}$. Then for any state ρ in $\mathcal{H}$, we have:*

$$\mathcal{R}_{\mathcal{C}}(\rho) = \bigvee_{i=0}^{d-1} \operatorname{supp}\left(\mathcal{E}^i(\rho)\right) = \operatorname{supp}\left(\sum_{i=0}^{d-1} \mathcal{E}^i(\rho)\right) \tag{9}$$

where $\mathcal{E}^i$ is the ith power of $\mathcal{E}$; that is, $\mathcal{E}^0 = \mathcal{I}$ (the identity operation in $\mathcal{H}$) and $\mathcal{E}^{i+1} = \mathcal{E} \circ \mathcal{E}^i$ for $i \geq 0$.

The reachable subspace $\mathcal{R}(\rho)$ can be viewed also as the least fixed point of quantum predicate transformer (see [30], Sect. 8.4) $\mathcal{T} : \mathcal{S}(\mathcal{H}) \to \mathcal{S}(\mathcal{H})$ defined by $\mathcal{T}(X) = \sup \rho \vee \mathcal{E}(X)$ for any $X \in \mathcal{S}(\mathcal{H})$.

4.2 Computing Reachable Subspaces

Based on Theorem 1, we can develop an algorithm for computing reachable subspaces in quantum Markov chain $\mathcal{C}$ using the tensor network representation of super-operator $\mathcal{E}$, with the help of the following:

Lemma 2. *Let $|\Psi\rangle = \sum_k |kk\rangle$ be the (unnormalised) maximally entangled state in $\mathcal{H} \otimes \mathcal{H}$. Then $(\mathcal{E}(A) \otimes I)|\Psi\rangle = M_{\mathcal{E}}(A \otimes I)|\Psi\rangle$, where I is the identity operator on $\mathcal{H}$.*

The basic idea of the algorithm is as follows. Define state $|\eta\rangle = \sum_{i=0}^{d-1} \mathcal{E}^i(\rho)$ in $\mathcal{H}$ and state $|\Phi\rangle = (\eta \otimes I)|\Psi\rangle$ in $\mathcal{H} \otimes \mathcal{H}$. Repeatedly using Lemma 2, we obtain:

$$|\Phi\rangle = \sum_{i=0}^{d-1} \left(\mathcal{E}^i(\rho) \otimes I\right)|\Psi\rangle = \sum_{i=0}^{d-1} M_{\mathcal{E}}^i(\rho \otimes I)|\Psi\rangle.$$

Thus, state $|\Phi\rangle$ can be computed by contracting the tensor network representations of $M_{\mathcal{E}}$, ρ and $|\Psi\rangle$. Finally, we can find the Schmidt decomposition of $|\Phi\rangle$: $|\Phi\rangle = \sum_j p_j |j\rangle \otimes |j'\rangle$, where $p_j > 0$ for all j. Then the reachable subspace $\mathcal{R}_{\mathcal{C}}(\rho) = \operatorname{span}\{|j\rangle\}$ is computed. Of course, the optimisation techniques for contracting tensor networks discussed in Sect. 3.4 can be applied here and combined with Lemma 1 when $\mathcal{E}$ comes from (sequential and parallel) compositions of smaller super-operators on subsystems.

5 Temporal Quantum Logic

Now let us move on to consider the verification problem for a more general class of properties of quantum circuits. To specify these properties, we define an assertion language for quantum circuits in this section. We choose to simply use Birkhoff-von Neumann quantum logic [2] for specifying static behaviour of quantum circuits. To specify their behaviour over time, however, we need to introduce a temporal extension of Birkhoff-von Neumann logic. Several other temporal logics have been defined in the literature [1,9,26,33,38] that are able to specify some sophisticated properties of quantum circuits than this logic. But we decide to adopt this simple temporal logic because its model checking can be much more efficiently implemented and may find practical applications in the early stages of quantum design automation.

5.1 Birkhoff-von Neumann Quantum Logic

Birkhoff-von Neumann logic is a *propositional logic* for reasoning about (static properties of) quantum systems. We assume an alphabet consisting of:

- a set AP of atomic propositions, ranged over by metavariables $X, X_1, X_2, ...$; and
- propositional connectives $\neg$ (negation) and $\wedge$ (conjunction).

Given a Hilbert space $\mathcal{H}$ as the state space of the quantum circuit under consideration. We write $\mathcal{S}(\mathcal{H})$ for the set of its closed subspaces. It is well-known that $(\mathcal{S}(\mathcal{H}), \cap, \vee, \perp)$ is an orthomodular lattice with inclusion $\subseteq$ as its ordering, where $\cap, \vee$ and $\perp$ stand for intersection, join defined in Eq. (8), and orthocomplement, i.e. $X^\perp = \{|\psi\rangle : |\psi\rangle \text{ is orthogonal to all } |\varphi\rangle \in X\}$. Then atomic propositions are interpreted as subspaces of $\mathcal{H}$, i.e. elements of $\mathcal{S}(\mathcal{H})$, and connectives $\neg, \wedge$ are interpreted as $\perp$ and $\cap$, respectively. For each logical formula A, its semantics $[\![A]\!]$ is a subspace of $\mathcal{H}$, meaning that the circuit's current state is within the region $[\![A]\!]$, and $\neg A$ indicates that the probability that the circuit's state enters the region $[\![A]\!]$ is zero. We can define $\vee$ (disjunction) by $A \vee B := \neg(\neg A \wedge \neg B)$, and it is easy to see that $[\![A \vee B]\!] = [\![A]\!] \vee [\![B]\!]$ with the symbol $\vee$ in the right-hand side being join. Moreover, satisfaction of a proposition A by a pure state $|\psi\rangle$ or a mixed state ρ is simply defined as follows:

$$\varphi \models A \text{ iff } \varphi \in [\![A]\!], \qquad \rho \models A \text{ iff } \mathrm{supp}(\rho) \subseteq [\![A]\!]. \tag{10}$$

5.2 Computation Tree Quantum Logic

A temporal extension of quantum logic can be naturally defined. For the limitation of space, we only consider computation tree quantum logic CTQL. Its syntax is the same as that of classical computation tree logic CTL:

- State formulas: $\Phi ::= A \mid \exists\varphi \mid \forall\varphi \mid \neg\Phi \mid \Phi_1 \wedge \Phi_2$
- Path formulas: $\varphi ::= O\Phi \mid \Phi_1 U \Phi_2$

except that A stands here for a propositional formula in Birkhoff-von Neumann quantum logic rather than a classical (two-valued) proposition.

Simulation-Based Semantics: We define the semantics of CTQL with the following *design decision*: our verification of quantum circuits will be done by simulation on a classical computer. Therefore, no actual quantum measurement is performed for checking whether a quantum state $|\varphi\rangle$ or ρ is in a subspace X, i.e. $|\varphi\rangle \models X$ or $\rho \models X$ according to Eq. (10), and thus no quantum state decaying happens. Let $\mathcal{S} = \langle \mathcal{H}, L, l_0, \mathcal{T} \rangle$ be a QTS. Then a configuration of $\mathcal{S}$ is a pair (l, ρ), where $l \in L$ is a location and ρ is a quantum state in $\mathcal{H}$. We write $\mathcal{C}(\mathcal{S})$ for the set of configurations of $\mathcal{S}$. A sequence $\pi = (l_1, \rho_1)(l_2, \rho_2) \cdots (l_{i-1}, \rho_{i-1})(l_i, \rho_i) \cdots$ of configurations is a path in $\mathcal{S}$ if there exists a sequence $l_1 \xrightarrow{\mathcal{E}_1} l_2 \xrightarrow{\mathcal{E}_2} ... \xrightarrow{\mathcal{E}_{i-1}} l_i \xrightarrow{\mathcal{E}_i} \cdots$ of transitions such that $\rho_{i+1} = \mathcal{E}_i(\rho_i)$ for all i. We often write $\pi[i] = (l_{i+1}, \rho_{i+11})$ for $i \geq 1$. Then the satisfaction relation in CTL can be straightforwardly generalised to CTQL:

Definition 4. *1. Satisfaction* $(l, \rho) \models \Phi$ *for state formulas is defined as follows:*

(a) $(l, \rho) \models A$ *iff* $\mathrm{supp}(\rho) \subseteq [\![A]\!]$*;*

(b) $(l, \rho) \models \exists\varphi$ *iff* $\pi \models \varphi$ *for some path* π *starting in* (l, ρ)*;*

(c) $(l, \rho) \models \forall\varphi$ *iff* $\pi \models \varphi$ *for all paths* π *starting in* (l, ρ)*;*

(d) $(l, \rho) \models \neg\Phi$ *iff* $\rho \not\models \Phi$*;*

(e) $(l, \rho) \models \Phi_1 \wedge \Phi_2$ *iff* $(l, \rho) \models \Phi_1$ *and* $(l, \rho) \models \Phi_2$*.*

2. Satisfaction $\pi \models \varphi$ *for path formulas is defined as follows:*

(a) $\pi \models O\Phi$ *iff* $\pi[1] \models \Phi$*;*

(b) $\pi \models \Phi_1 U \Phi_2$ *iff there exists* $i \geq 0$ *such that* $\pi[i] \models \Phi_2$ *and* $\pi[j] \models \Phi_1$ *for all* $0 \leq j < i$*.*

3. We say that $\mathcal{S}$ *with initial state* ρ *satisfies* Φ*, written* $(\mathcal{S}, \rho) \models \Phi$*, if* $(l_0, \rho) \models \Phi$*.*

Remark 2. The above simulation-based semantics is fundamentally different from the measurement-based semantics of quantum temporal logics considered in the previous literature where the system's state is disturbed by a measurement, and the system's next step starts from the post-measurement state.

6 Model Checking Quantum Circuits

In this section, we show how model checking can be used in verification of the properties of quantum circuits specified in temporal logic CQTL introduced in the last section.

6.1 CTQL Model Checking

Indeed, classical CTL model checking techniques can be adapted to solve the following:

- **CTQL model checking problem**: Given a QTS $\mathcal{S} = \langle \mathcal{H}, L, l_0, \mathcal{T} \rangle$, an initial state ρ and a CTQL state formula Φ. Check $(\mathcal{S}, \rho) \models \Phi$?

The basic idea is to construct a classical transition system $\overline{\mathcal{S}}_\rho$ from a QTS with an initial state ρ so that the above CTQL model checking problem is reduced to a CTL model checking problem. We construct $\overline{\mathcal{S}}_\rho = \langle \mathcal{C}(\mathcal{S})_\rho, \Rightarrow, (l_0, \rho), L \rangle$ as follows:

- Transition relation $\Rightarrow$ between configurations $(l, \rho), (l', \rho') \in \mathcal{C}(\mathcal{S})$ is defined by
$$(l, \rho) \Rightarrow (l', \rho') \text{ iff for some } \mathcal{E} : \ l \xrightarrow{\mathcal{E}} l' \text{ and } \rho' = \mathcal{E}(\rho); \tag{11}$$
- We define $\mathcal{C}(\mathcal{S})_\rho$ as the set of configurations reachable from (l_0, ρ) through $\Rightarrow$;
- Configuration (l_0, ρ) is defined as the initial state of $\overline{\mathcal{S}}_\rho$;
- Propositional symbols A in CTQL are interpreted as propositions in Birkhoff-von Neumann quantum logic and thus their semantics $[\![A]\!]$ are subspaces of $\mathcal{H}$. However, in CTL for classical transition system $\overline{\mathcal{S}}_\rho$, they are considered as classical two-valued propositions, and labelling function L interprets A as follows: for each $(l, \sigma) \in \mathcal{C}(\mathcal{S})_\rho$,
$$A \in L(l, \sigma), \text{ i.e. } (l, \sigma) \models A \text{ iff } \mathrm{supp}(\rho) \subseteq [\![A]\!]. \tag{12}$$

The following simple lemma establishes a connection between CTQL for a QTS $\mathcal{S}$ and CTL for the classical transition system $\overline{\mathcal{S}}_\rho$ defined from $\mathcal{S}$ with an initial state ρ.

Lemma 3. *For any CTQL state formula $\varPhi$, any QTS $\mathcal{S}$ and any quantum state ρ in $\mathcal{S}$,*

$$(\mathcal{S}, \rho) \models \varPhi \text{ iff } \overline{\mathcal{S}}_\rho \models \varPhi. \tag{13}$$

Note that in the left-hand side of (13), $\varPhi$ is treated as a CTQL formula, but in the right-hand side, it is seen as a CTL formula in which atomic propositions A are interpreted by labelling function L defined in Eq. (12).

Based on Lemma 3, whenever $\mathcal{C}(\mathcal{S})_\rho$ is finite, then CTL model checking algorithms together with computations of (11) and (12) can be used to check whether $\overline{\mathcal{S}}_\rho \models \varPhi$ or not. However, it is possible that $\mathcal{C}(\mathcal{S})_\rho$ is infinite. In this case, we can apply bounded model checking to check the configurations reachable from (l_0, ρ) through $\leq k$ steps.

6.2 Assertion-Based Verification of Quantum Circuits

The above discussion indicates that assertions about quantum circuits written in CTQL can be verified by CTL model checking with some extra computations. It is well-known that a major practical hurdle in model checking applied to verifying classical circuits is the state space explosion problem. As one can imagine, this problem unavoidably occurs in the case of quantum circuits. The tensor network representation of quantum circuits discussed in Sect. 3, together with various partitioning techniques for quantum transition systems defined in Sect. 2.5 that exploit the locality in the circuits, can be a remedy to this issue. More explicitly, it is very helpful in computing reachable configurations $\mathcal{C}(\mathcal{S})_\rho$ and the labelling function (12). The symbolic representation of quantum circuits using matrix-valued Boolean expressions proposed in [32] should also be useful.

7 Conclusion

In this paper, we presented a framework for assertion-based verification of quantum circuits by model checking with the help of tensor networks. The verified properties are *qualitative* assertions written in a temporal extension of Birkhoff-von Neumann quantum logic. This modest aim of verifying only qualitative assertions is identified mainly for the reason that the verification algorithm can be more efficiently implemented and thus is actually useful in short-term practical applications. To check *quantitative* assertions (with probabilities) for quantum systems, some techniques have been developed in [9,31,36,38], but the involved computation are overwhelming. To remedy this seemingly inevitable inefficiency of verifying quantum circuits on classical computers, we also tried to develop quantum algorithms for model checking quantum systems [13].

Acknowledgment. This work has been partly supported by the National Key R&D Program of China (2018YFA0306701), the Australian Research Council (DP210102449) and the National Natural Science Foundation of China (61832015).

References

1. Baltazar, P., Chadha, R., Mateus, P.: Quantum computation tree logic: model checking and complete calculus. Int. J. Quant. Inf. **6**, 219–36 (2008)
2. Birkhoff, G., von Neumann, J.: The logic of quantum mechanics. Ann. Math. **37**, 823–843 (1936)
3. Bocharov, A., Roetteler, M., Svore, K.M.: Efficient synthesis of universal repeat-until-success quantum circuits. Phys. Rev. Lett. **114**, 080502 (2015)
4. Boule, M., Chenard, J.-S., Zilic, Z.: Assertion checkers in verification, silicon debug and in-field diagnosis. In: 8th IEEE International Symposium on Quality Electronic Design, pp. 613–620 (2007)
5. Burch, J.R., Clarke, E.D., Long, D.E., McMillan, K.L., Dill, D.L.: Symbolic model checking for sequential circuit verification. IEEE Trans. Comput. Aided Des. Integr. Circ. Syst. **13**, 401–424 (1994)
6. Burgholzer, L., Wille, R.: Advanced equivalence checking for quantum circuits. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **40**(9), 1810–1824 (2021)
7. Corcoles, A.D., et al.: Exploiting dynamic quantum circuits in a quantum algorithm with superconducting qubits, arXiv: 2102:01682
8. Davidson, T.A., Gay, S.J., Mlnarik, H., Nagarajan, R., Papanikolaou, N.: Model checking for communicating quantum processes. Int. J. Unconv. Comput. **8**, 73–98 (2011)
9. Feng, Y., Yu, N.K., Ying, M.S.: Model checking quantum Markov chains. J. Comput. Syst. Sci. **79**, 1181–1198 (2013)
10. D Foster, H., Marschner, E.: Assertion-based verification, In: Lavagno, L., Martin, G.M., Markov, I.L., Scheffer, L.K. (eds.) Electronic Design Automation for IC System Design, Verification, and Testing, pp. 441-460. CRC Press (2016)
11. Gay, S.J., Nagarajan, R., Papanikolaou, N.: QMC: a model checker for quantum systems. In: Gupta, A., Malik, S. (eds.) CAV 2008. LNCS, vol. 5123, pp. 543–547. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-70545-1_51
12. Gough, J.E., James, M.R.: Quantum feedback network: Hamiltonian formulation. Commun. Math. Phys. **287**, 1109–1132 (2008). https://doi.org/10.1007/s00220-008-0698-8
13. Guan, J., Wang, Q.S., Ying, M.S.: An HHL-based algorithm for computing hitting probabilities of quantum random walks, Quantum Information & Computation (2021)
14. Gudder, S.: Quantum Markov chains. J. Math. Phys. **49**, 072105 (2008)
15. Häner, T., Steiger, D.S.: 0.5 petabyte simulation of a 45-qubit quantum circuit. In: Proceedings of the SC 2017, pp. 1–10 (2017)
16. Hong, X., Zhou, X.Z., Li, S.J., Feng, Y., Ying, M.S.: A tensor network based decision diagram for representation of quantum circuits, arXiv: 2009.02618
17. Huang, C.J.: et al.: Classical simulation of quantum supremacy circuits, arXiv:2005.06787
18. Kerckhoff, J., Nurdin, H.I., Pavlichin, D.S., Mabuchi, H.: Designing quantum memories with embedded control: photonic circuits for autonomous quantum error correction. Phys. Rev. Lett. **105**, 040502 (2010)

19. Li, R.L., Wu, B.J., Ying, M.S., Sun, X.M., Yang, G.W.: Quantum supremacy circuit simulation on Sunway TaihuLight. IEEE Trans. Parallel Distrib. Syst. **31**, 805–816 (2020)
20. Li, Y., Ying, M.: (Un)decidable problems about reachability of quantum systems. In: Baldan, P., Gorla, D. (eds.) CONCUR 2014. LNCS, vol. 8704, pp. 482–496. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44584-6_33
21. Pednault, E., et al.: Breaking the 49-qubit barrier in the simulation of quantum circuits, arXiv:1710.05867
22. Seiter, J., Soeken, M., Wille, R., Drechsler, R.: Property checking of quantum circuits using quantum multiple-valued decision diagrams. In: Glück, R., Yokoyama, T. (eds.) RC 2012. LNCS, vol. 7581, pp. 183–196. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36315-3_15
23. Tezak, N., Niederberger, A., Pavlichin, D.S., Sarma, G., Mabuchi, H.: Specification of photonic circuits using quantum hardware description language. Philos. Trans. R. Soc. A **370**, 5270–5290 (2012)
24. Viamontes, G.F., Markov, I.L., Hayes, J.P.: Improving gate-level simulation of quantum circuits. Quantum Inf. Process. **2**, 347–379 (2004). https://doi.org/10.1023/B:QINP.0000022725.70000.4a
25. Viamontes, G.F., Markov, I.L., Hayes, J.P.: Checking equivalence of quantum circuits and states. In: Proceedings of the ICCAD 2007, pp. 69–74 (2007)
26. Vigano, L., Volpe, M., Zorzi, M.: A branching distributed temporal logic for reasoning about entanglement-free quantum state transformations. Inf. Comput. **255**, 311–333 (2017)
27. Villalonga, B., et al.: A flexible high-performance simulator for verifying and benchmarking quantum circuits implemented on real hardware. NPJ Quantum Inf. **5**, 1–16 (2019). Art. no. 86
28. Wang ,Q.S., Ying, M.S.: Equivalence checking of sequential quantum circuits, arXiv:1811.07722
29. Ying, M.S.: Foundations of Quantum Programming. Morgan Kaufmann, Cambridge, MA, USA (2016)
30. Ying, M.S., Duan, R.Y., Feng, Y., Ji, Z.F.: Predicate transformer semantics of quantum programs. In: Mackie, I., Gay, S. (eds.) Semantic Techniques in Quantum Computation, pp. 311–360. Cambridge University Press (2010)
31. Ying, M.S., Feng, Y.: Model Checking Quantum Systems: Principles and Algorithms. Cambridge University Press, Cambridge, UK (2021)
32. Ying, M.S., Ji, Z.F.: Symbolic verification of quantum circuits, arXiv: 2010.03032
33. Ying, M.S., Li, Y.J., Yu, N.K., Feng, Y.: Model-checking linear-time properties of quantum systems. ACM Trans. Comput. Logic **15**, 1–31 (2014). Art. no. 22
34. Ying, M.S., Ying, S.G., Wu, X.D.: Invariants of quantum programs: characterisations and generation. In: POPL 2017, pp. 818–832 (2017)
35. Ying, M.S., Yu, N.K., Feng, Y., Duan, R.Y.: Verification of quantum programs. Sci. Comput. Program. **78**, 1679–1700 (2013)
36. Ying, S., Feng, Y., Yu, N., Ying, M.: Reachability probabilities of Quantum Markov chains. In: D'Argenio, P.R., Melgratti, H. (eds.) CONCUR 2013. LNCS, vol. 8052, pp. 334–348. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40184-8_24
37. Yu, N., Ying, M.: Reachability and termination analysis of concurrent quantum programs. In: Koutny, M., Ulidowski, I. (eds.) CONCUR 2012. LNCS, vol. 7454, pp. 69–83. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32940-1_7
38. Yu, N.K.: Quantum temporal logic, arXiv:1908.00158